Version 2021-5-31

Riemannian distances are invariant to forward and inverse mappings in MEG/EEG

Kensuke Sekihara

Introduction

The arguments in this note are inspired from [1] (maybe my interpretation of the arguments in [1]).

One of large application fields of MEG/EEG is the inference of brain's internal states (such as subject's will/intention or subject's personal preference) based only on MEG/EEG signals. For such applications, we deal with the signal classification problems, and quite often, second-order statistics (such as the covariance, correlation, and coherence) are used. (In this note, it is assumed that covariance matrices are used.)

Particulary when MEG is used, there are two choices , which are using the sensor-space covariance or using the source-space covariance. In most cases, such choices have been made empirically, and so far very little have been reported on their rigorous comparisons as far as I know. This note proves the theoretical equivalence between the uses of these two-types of covariance matrices for signal classification, if the Riemannian distance, called the affine invariant Riemannian metric (AIRM), are used as a measure for their similaity or de-similarity.

Data model and definitions

MEG data model is

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{\varepsilon}. \tag{1}$$

Here, $\boldsymbol{y}$, $\boldsymbol{\varepsilon} \in \mathbb{R}^{M\times 1}$, $\boldsymbol{H} \in \mathbb{R}^{M\times Q}$, and $\boldsymbol{x} \in \mathbb{R}^{Q\times 1}$, where M is the number of sensors and Q is the number of sources. (We assume the scalar lead field.) We can say (in a manner modern mathematics prefers) that there is a (surjective) map $\boldsymbol{H}$ from the source space to the sensor space[1]. Inverse algorithms try to obtain inverse maps from the sensor space to the source space. Once an appropriate inverse map is obtained, we perform the source localization,

$$\widehat{\boldsymbol{x}} = \boldsymbol{W}\boldsymbol{y}.$$

Here, $\boldsymbol{W}$ is a map from the source space to the sensor space. The relationship between covariance matrices in the source and sensor spaces is

$$\boldsymbol{R}_y = \boldsymbol{H}\boldsymbol{R}_x\boldsymbol{H}^T + \boldsymbol{R}_\varepsilon,$$

where $\boldsymbol{R}_x$, $\boldsymbol{R}_x$, and $\boldsymbol{R}_\varepsilon$ respectively denote the source, sensor, and noise covariance matries

[1] The source and sensor spaces are the vector spaces of the source vector $\boldsymbol{x}$ and the observation vector $\boldsymbol{y}$.

A simple example

We use an example for classification between two conditions. Sensor-data covariance matrices corresponding to conditions A and B are given by $\boldsymbol{R}_y^A$ and $\boldsymbol{R}_y^B$. The essential step for the signal classification is quantitatively assess the difference between $\boldsymbol{R}_y^A$ and $\boldsymbol{R}_y^B$; the difference is called the distance between $\boldsymbol{R}_y^A$ and $\boldsymbol{R}_y^B$. In this note, this distance is computed as the geodesic distance on the SPD manifold, $\delta(\boldsymbol{R}_y^A, \boldsymbol{R}_y^B)$, derived using the affine invariant Riemannian metric (AIRM). The AIRM-base geodesic distance is given by

$$\delta(\boldsymbol{R}_y^A, \boldsymbol{R}_y^B) = \|\mathrm{Log}\boldsymbol{R}_y^A - \mathrm{Log}\boldsymbol{R}_y^B\| = \|\mathrm{Log}\left[(\boldsymbol{R}_y^B)^{-1/2}\boldsymbol{R}_y^A(\boldsymbol{R}_y^B)^{-1/2}\right]\|.$$

Isometric mapping within the SPD manifold

Let us consider a map $F_X : \mathcal{P}(n) \to \mathcal{P}(n)$:

$$F_X(\boldsymbol{P}) = \boldsymbol{X}^T\boldsymbol{P}\boldsymbol{X} \tag{2}$$

where $\boldsymbol{X}$ is a non-singular matrix. This map is an isometory, and preserves the length of a path in $\mathcal{P}(n)$, such that

$$\delta(F_X(\boldsymbol{A}), F_X(\boldsymbol{B})) = \delta(\boldsymbol{A}, \boldsymbol{B}). \tag{3}$$

That is, the geodesic distance between $\boldsymbol{A}$ and $\boldsymbol{B}$ is equal to that between $F_X(\boldsymbol{A})$ and $F_X(\boldsymbol{B})$.

Modified data model

To make use of this isometry, we approximate the sensor noise term as

$$\boldsymbol{\varepsilon} = \boldsymbol{A}\boldsymbol{u}, \tag{4}$$

where $\boldsymbol{A} \in \mathbb{R}^{M\times(M-Q)}$ is a mixing matrix, and $\boldsymbol{u} \in \mathbb{R}^{(M-Q)\times 1}$ indicates factors. That is, the sensor noise is expressed as the summation of $M - Q$ factor activities. Since the sensor noise should be expressed as M factor activities, the assumption in Eq (4) is obviously wrong. However, if $M \gg Q$, Eq (4) may be an acceptable assumption.

The MEG data model then is rewritten as

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{A}\boldsymbol{u} = [\boldsymbol{H}, \boldsymbol{A}]\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{u}\end{bmatrix} = \boldsymbol{F}\boldsymbol{z}. \tag{5}$$

Here, $\boldsymbol{F}$ is an $M \times M$ non-singular square matrix, and $\boldsymbol{z} \in \mathbb{R}^{M\times 1}$, which is a vector containing source and noise activities.

Geodesic distances are the same between source and sensor spaces

The covariance matrix of $\boldsymbol{z}$ is defined as $\boldsymbol{R}_z$:

$$\boldsymbol{R}_y = \boldsymbol{F}\boldsymbol{R}_z\boldsymbol{F}^T.$$

Since $\boldsymbol{F}$ is a non-singular matrix, the above equation indicates an isometric mapping (Eq. (2)) from $\boldsymbol{R}_z$ to $\boldsymbol{R}_y$. Therefore, using Eq. (3), we have

$$\delta(\boldsymbol{R}_y^A, \boldsymbol{R}_y^B) = \delta(\boldsymbol{F}\boldsymbol{R}_z^A\boldsymbol{F}^T, \boldsymbol{F}\boldsymbol{R}_z^B\boldsymbol{F}^T) = \delta(\boldsymbol{R}_z^A, \boldsymbol{R}_z^B). \tag{6}$$

Here, $\boldsymbol{R}_z$ is expressed as

$$\boldsymbol{R}_z = E(\boldsymbol{z}\boldsymbol{z}^T) = E\left[\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{bmatrix}[\boldsymbol{x}^T, \boldsymbol{u}^T]\right] = \begin{bmatrix} E(\boldsymbol{x}\boldsymbol{x}^T) & E(\boldsymbol{x}\boldsymbol{u}^T) \\ E(\boldsymbol{u}\boldsymbol{x}^T) & E(\boldsymbol{u}\boldsymbol{u}^T) \end{bmatrix} = \begin{bmatrix} E(\boldsymbol{x}\boldsymbol{x}^T) & \boldsymbol{0} \\ \boldsymbol{0} & E(\boldsymbol{u}\boldsymbol{u}^T) \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_x & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_u \end{bmatrix}. \tag{7}$$

Therefore, the geodesic distance is expressed as

$$\begin{aligned}
\delta(\boldsymbol{R}_z^A, \boldsymbol{R}_z^B) &= \|\mathrm{Log}\begin{bmatrix} \boldsymbol{R}_x^A & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_u \end{bmatrix} - \mathrm{Log}\begin{bmatrix} \boldsymbol{R}_x^B & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_u \end{bmatrix}\| \\
&= \|\mathrm{Log}\left(\begin{bmatrix} \boldsymbol{R}_x^B & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_u \end{bmatrix}^{-1/2}\begin{bmatrix} \boldsymbol{R}_x^A & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_u \end{bmatrix}\begin{bmatrix} \boldsymbol{R}_x^B & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_u \end{bmatrix}^{-1/2}\right)\| \\
&= \|\mathrm{Log}\left(\begin{bmatrix} (\boldsymbol{R}_x^B)^{-1/2}\boldsymbol{R}_x^A & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_u^{1/2} \end{bmatrix}\begin{bmatrix} \boldsymbol{R}_x^B & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_u \end{bmatrix}^{-1/2}\right)\| = \|\mathrm{Log}\begin{bmatrix} (\boldsymbol{R}_x^B)^{-1/2}\boldsymbol{R}_x^A(\boldsymbol{R}_x^B)^{-1/2} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}\| \\
&= \|\mathrm{Log}\left[(\boldsymbol{R}_x^B)^{-1/2}\boldsymbol{R}_x^A(\boldsymbol{R}_x^B)^{-1/2}\right]\|.
\end{aligned} \tag{8}$$

The proof that the equation:

$$\|\mathrm{Log}\begin{bmatrix} (\boldsymbol{R}_x^B)^{-1/2}\boldsymbol{R}_x^A(\boldsymbol{R}_x^B)^{-1/2} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}\| = \|\mathrm{Log}\left[(\boldsymbol{R}_x^B)^{-1/2}\boldsymbol{R}_x^A(\boldsymbol{R}_x^B)^{-1/2}\right]\| \tag{9}$$

holds will be shown in Appendix.

On the other hand, we have

$$\delta(\boldsymbol{R}_x^A, \boldsymbol{R}_x^B) = \|\mathrm{Log}\boldsymbol{R}_x^A - \mathrm{Log}\boldsymbol{R}_x^B\| = \|\mathrm{Log}\left[(\boldsymbol{R}_x^B)^{-1/2}\boldsymbol{R}_x^A(\boldsymbol{R}_x^B)^{-1/2}\right]\|. \tag{10}$$

Therefore, we have proved that the relationship

$$\delta(\boldsymbol{R}_z^A, \boldsymbol{R}_z^B) = \delta(\boldsymbol{R}_x^A, \boldsymbol{R}_x^B) \tag{11}$$

holds. Combining Eqs. (6) and (11) we finally have

$$\delta(\boldsymbol{R}_y^A, \boldsymbol{R}_y^B) = \delta(\boldsymbol{R}_x^A, \boldsymbol{R}_x^B). \tag{12}$$

The above equation implies that the geodesic distance obtained from AIRM is the same between the sensor and source covariance matrices.

Summary

We have shown that the AIRM-based Riemannian diatnce is invariant to the forward and inverse mappings between the source and sensor spaces in MEG/EEG.

References

[1] Sabbagh, D., Ablin, P., Varoquaux, G., Gramfort, A., Engemann, D. A. (2019). Manifold-regression to predict from MEG/EEG brain signals without source modeling. arXiv preprint arXiv:1906.02687.

A Proof of Eq.(9)

This appendix proves that the relationship,

$$\|\mathrm{Log}\begin{bmatrix} \boldsymbol{P} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix}\| = \|\mathrm{Log}\boldsymbol{P}\|$$

holds where $\boldsymbol{P} \in \mathbb{R}^{Q\times Q}$ is a symmetric positive definite (SPD) matrix, and $\boldsymbol{I} \in \mathbb{R}^{(M-Q)\times(M-Q)}$ is an identity matrix.

Since $\boldsymbol{P}$ is a SPD, its eigen-decomposition is given by

$$\boldsymbol{P} = \sum_{j=1}^{Q} \lambda_j \boldsymbol{d}_j \boldsymbol{d}_j^T,$$

where λ_j is a positive eigenvalue, and $[\boldsymbol{d}_1, \ldots, \boldsymbol{d}_Q]$ is a $Q \times Q$ orthogonal matrix. Also, the identity matrix $\boldsymbol{I}$ can be expressed as

$$\boldsymbol{I} = \sum_{j=1}^{M-Q} \boldsymbol{e}_j \boldsymbol{e}_j^T,$$

where $[\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{M-Q}]$ is an $(M-Q) \times (M-Q)$ orthogonal matrix. Then, we have

$$\begin{bmatrix} \boldsymbol{P} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{Q} \lambda_j \boldsymbol{d}_j \boldsymbol{d}_j^T & \mathbf{0} \\ \mathbf{0} & \sum_{j=1}^{M-Q} \boldsymbol{e}_j \boldsymbol{e}_j^T \end{bmatrix} \tag{13}$$

Let us define column vectors $\boldsymbol{\alpha}_j \in \mathbb{R}^{M\times 1}$ $(j = 1, \ldots, M)$ such that

$$\boldsymbol{\alpha}_j = \begin{bmatrix} \boldsymbol{d}_j \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{for} \quad j = 1, \ldots, Q, \quad \text{and} \quad \boldsymbol{\alpha}_{Q+j} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \boldsymbol{e}_j \end{bmatrix} \quad \text{for} \quad j = 1, \ldots, M-Q,$$

and define an $M \times M$ matrix $\boldsymbol{S}$ such that

$$\boldsymbol{S} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_Q, \boldsymbol{\alpha}_{Q+1}, \ldots, \boldsymbol{\alpha}_M] \begin{bmatrix} \lambda_1 & 0 & \cdots & \cdot & \cdot & \cdots & 0 \\ 0 & \lambda_2 & \cdots & \cdot & \cdot & \cdot & \cdot \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdot & \cdots & \lambda_Q & 0 & \cdots & 0 \\ \cdot & \cdot & \cdots & \cdot & 1 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdot & \cdots & \cdot & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_1^T \\ \vdots \\ \boldsymbol{\alpha}_Q^T \\ \boldsymbol{\alpha}_{Q+1}^T \\ \vdots \\ \boldsymbol{\alpha}_M^T \end{bmatrix}$$

$$= \sum_{j=1}^{Q} \lambda_j \boldsymbol{\alpha}_j \boldsymbol{\alpha}_j^T + \sum_{j=Q+1}^{M} \boldsymbol{\alpha}_j \boldsymbol{\alpha}_j^T \tag{14}$$

In the right-most side of the above equation, the first term is expressed as

$$\sum_{j=1}^{Q} \lambda_j \boldsymbol{\alpha}_j \boldsymbol{\alpha}_j^T = \sum_{j=1}^{Q} \lambda_j \begin{bmatrix} \boldsymbol{d}_j \\ 0 \\ \vdots \\ 0 \end{bmatrix} [\boldsymbol{d}_j^T, 0, \ldots, 0] = \begin{bmatrix} \sum_{j=1}^{Q} \lambda_j \boldsymbol{d}_j \boldsymbol{d}_j^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

The second term is expressed as

$$\sum_{j=Q+1}^{M} \lambda_j \boldsymbol{\alpha}_j \boldsymbol{\alpha}_j^T = \sum_{j=Q+1}^{M} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \boldsymbol{e}_{(j-Q)} \end{bmatrix} [0, \ldots, 0, \boldsymbol{e}_{(j-Q)}^T] = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sum_{j=1}^{M-Q} \boldsymbol{e}_j \boldsymbol{e}_j^T \end{bmatrix}$$

Therefore, we get

$$\boldsymbol{S} = \begin{bmatrix} \sum_{j=1}^{Q} \lambda_j \boldsymbol{d}_j \boldsymbol{d}_j^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sum_{j=1}^{M-Q} \boldsymbol{e}_j \boldsymbol{e}_j^T \end{bmatrix} = \begin{bmatrix} \boldsymbol{P} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix}. \tag{15}$$

Eq. (14) is turned out to be the eigendecomposition of $\boldsymbol{S}$. Since we have

$$\begin{aligned} \mathrm{Log} \begin{bmatrix} \boldsymbol{P} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix} &= [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_Q, \boldsymbol{\alpha}_{Q+1}, \ldots, \boldsymbol{\alpha}_M] \begin{bmatrix} \log \lambda_1 & 0 & \cdots & \cdot & \cdot & \cdots & 0 \\ 0 & \log \lambda_2 & \cdots & \cdot & \cdot & \cdot & \cdot \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdot & \cdots & \log \lambda_Q & 0 & \cdots & 0 \\ \cdot & \cdot & \cdots & \cdot & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdot & \cdots & \cdot & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_1^T \\ \vdots \\ \boldsymbol{\alpha}_Q^T \\ \boldsymbol{\alpha}_{Q+1}^T \\ \vdots \\ \boldsymbol{\alpha}_M^T \end{bmatrix} \\ &= [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_Q] \begin{bmatrix} \log \lambda_1 & 0 & \cdots & 0 \\ 0 & \log \lambda_2 & \cdots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & \log \lambda_Q \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_1^T \\ \vdots \\ \boldsymbol{\alpha}_Q^T \end{bmatrix}, \end{aligned} \tag{16}$$

we finally get

$$\|\mathrm{Log} \begin{bmatrix} \boldsymbol{P} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix} \| = \| \begin{bmatrix} \mathrm{Log}\boldsymbol{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \| = \|\mathrm{Log}\boldsymbol{P}\|. \tag{17}$$

B Appendix: properties of matrix exponential function

For a matrix $\boldsymbol{X}$ ($\boldsymbol{X} \in \mathbb{M}(n)$), the matrix exponential function $e^{\boldsymbol{X}}$ is defined such that

$$e^{\boldsymbol{X}} = \sum_{k=0}^{\infty} \frac{1}{k!} \boldsymbol{X}^k. \tag{18}$$

The matrix exponential function has following properties:

1. $e^{\mathbf{0}} = \boldsymbol{I}$
2. If $\boldsymbol{A}$ is a diagonal matrix, i.e., $\boldsymbol{A} = \text{diag}(\lambda_1, \ldots, \lambda_n)$, then, $e^{\boldsymbol{A}} = \text{diag}(e^{\lambda_1}, \ldots, e^{\lambda_n})$.
3. If $\boldsymbol{AB} = \boldsymbol{BA}$ (if $\boldsymbol{A}$ and $\boldsymbol{B}$ commute), then $e^{\boldsymbol{A}} e^{\boldsymbol{B}} = e^{\boldsymbol{AB}}$.
4. $e^{a\boldsymbol{A}} e^{b\boldsymbol{A}} = e^{(a+b)\boldsymbol{A}}$ because $a\boldsymbol{A}$ and $b\boldsymbol{A}$ commute.
5. $e^{\boldsymbol{A}} e^{-\boldsymbol{A}} = \boldsymbol{I}$ because setting $a = 1$ and $b = -1$ in the above equation.
6. $e^{-\boldsymbol{A}} = (e^{\boldsymbol{A}})^{-1}$.
7. If $\boldsymbol{A} = \boldsymbol{P\Lambda P}^{-1}$ and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_n)$,

$$e^{\boldsymbol{A}} = \boldsymbol{P} \, \text{diag}(e^{\lambda_1}, \ldots, e^{\lambda_n}) \boldsymbol{P}^{-1},$$

   holds, because $(\boldsymbol{P\Lambda P}^{-1})^k = \boldsymbol{P\Lambda}^k \boldsymbol{P}^{-1}$ holds.
8. If $\boldsymbol{A} = \boldsymbol{U\Lambda U}^T$ and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_n)$,

$$e^{\boldsymbol{A}} = \boldsymbol{U} \, \text{diag}(e^{\lambda_1}, \ldots, e^{\lambda_n}) \boldsymbol{U}^T$$

   holds, because $(\boldsymbol{U\Lambda U}^T)^k = \boldsymbol{U\Lambda}^k \boldsymbol{U}^T$ holds.
9. $\frac{d}{dt} e^{t\boldsymbol{A}} = \boldsymbol{A} e^{t\boldsymbol{A}}$.
10. $\det(e^{\boldsymbol{A}}) = e^{\text{tr}(\boldsymbol{A})}$.
11. Matrix Log function: For $\boldsymbol{A} \in GL(n)$, matrix Log function is defined as

$$\text{Log}\boldsymbol{A} = -\sum_{k=1}^{\infty} \frac{(\boldsymbol{I} - \boldsymbol{A})^k}{k}.$$

   Actually, If $\boldsymbol{A} = \boldsymbol{P\Lambda P}^{-1}$ and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_n)$, the matrix log function is computed using

$$\text{Log}\boldsymbol{A} = \boldsymbol{P} \, \text{diag} \left[\log(\lambda_1), \ldots, \log(\lambda_n)\right] \boldsymbol{P}^{-1}.$$